IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

Applicant:	Kosche et al.	
App. No.:	10/050,358	Con. No.: 5734
Filed:	January 16, 2002	Art Unit: 2191
Title:	TECHNIQUE FOR ASSOCIATING INSTRUCTIONS WITH EXECUTION EVENTS	Examiner: Wang, Rongfa

AMENDMENT AND RESPONSE TO FINAL OFFICE ACTION

MAIL STOP AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

In response to the final Office action dated April 18, 2007, please consider the following remarks and amend the above-identified application as follows:

Amendments to the Claims begin on page 2 of this paper.

Remarks begin on page 9 of this paper.

Amendments to the Claims:

1. (Currently Amended) A method of profiling code for an execution environment in which latency exists between an execution event and detection thereof, the method comprising:

identifying an ambiguity creating location in the code, the ambiguity creating location being an operation reachable from a plurality of program execution paths;

executing the code on a processor;

detecting the execution event; [[and]]

backtracking a displacement ~~within an unambiguous skid region~~ from a detection point in the code coinciding with the detection of the execution event to a preceding operation;

determining that the ambiguity creating location is not encountered while backtracking; and[[,]]

wherein the displacement is based, at least in part, on a type of operation appropriate to have triggered the execution event.

2. (Cancelled)

3. (Currently Amended) The method of claim 1 further comprising,

if [[an]] the ambiguity creating location is encountered while backtracking, either ignoring the execution event or bridging the ambiguity creating location.

4. (Cancelled)

5. (Currently Amended) The method of claim 1, wherein the ~~unambiguous skid region does not include an ambiguity creating location and wherein the~~ ambiguity creating location is one of:

a jump target location;

an indirect branch target location;

a branch target location;

entry point location;

a trap handler location; and

an interrupt handler location.

6. (Original) The method of claim 1, wherein the preceding operation corresponds to a load instruction; and

wherein the execution event is a cache miss.

7. (Original) The method of claim 1, wherein the preceding operation corresponds to a memory access instruction; and

wherein the execution event is either a hit or miss at a level in a memory hierarchy.

8. (Previously Presented) The method of claim 1, wherein the execution event is either an overflow or an underflow of a hardware counter.

9. (Previously Presented) The method of claim 1, wherein the execution event triggers either an overflow or an underflow of a hardware counter that is itself detected.

10. (Original) The method of claim 1, wherein the latency includes that associated with delivery of a trap.

11. (Original) The method of claim 1, wherein the latency includes that associated with delivery of a counter overflow event signal.

12. (Original) the method of claim 1, wherein the latency is associated with pipeline execution skid.

13. (Original) the method of claim 1, wherein the latency is associated with completion of in-flight operations.

14. (Original) The method of claim 1, embodied in a computer program product.

15. (Original) the method of claim 1, embodied in a least one of:

a profiling tool;

a code optimizer; and

a runtime library.

16. (Original) The method of claim 1, employed in combination with a compiler that pads the code with one or more padding operations to absorb at least some instances of the latency.

17. (Original) The method of claim 16, wherein the padding operations are not themselves associated with the execution event.

18. (Currently Amended) The method of claim 16, wherein each padding operation is not [[an]] the ambiguity creating location.

19. (Currently Amended) A method of identifying operations associated with execution events of a processor, the method comprising:

identifying a target operation and an ambiguity creating location within an execution sequence of operations of the processor;

from a point in [[an]] the execution sequence of the operations of the processor, the point coinciding with an execution event, backtracking through the operations toward a particular operation that precedes the coinciding point by a displacement ~~within an unambiguous skid region~~, wherein the displacement is based, at least in part on a type of operation appropriate for triggering the execution event; [[and]]

determining that the ambiguity creating location is not encountered while backtracking; and

associating the execution event with the particular operation.

20. (Previously Presented) The method of claim 19, further comprising:

executing the sequence of operations on the processor; and

detecting the execution event.

21. (Previously Presented) The method of claim 19, wherein the operations are instructions executable on the processor; and

wherein the particular operation is a particular one of the instructions that triggers the execution event.

22. (Previously Presented) The method of claim 19, wherein the operations correspond to instructions of program code.

23. (Previously Presented) The method of claim 19, wherein the execution event is an exception triggering execution of the particular operation.

24. (Original) The method of claim 19, wherein the execution event is a cache miss.

25. (Previously Presented) The method of claim 19, wherein the displacement is based, at least in part, on a trap delivery delay.

26. (Previously Presented) The method of claim 19, wherein the execution event triggers a hardware event and the displacement is based, at least in part, on delivery of a signal associated therewith.

27. (Original) The method of claim 26, wherein the hardware event is either underflow or overflow of a counter associated with the execution event.

28. (Original) The method of claim 19, wherein the execution event is either underflow or overflow of a counter.

29. (Previously Presented) The method of claim 19, wherein one or more instances of an intervening target of a control transfer is identified in the execution sequence of operations to facilitate the backtracking.

30. (Original) The method of claim 29, wherein at least some of the instances of intervening control transfer targets are resolved using branch history information.

31. (Currently Amended) The method of claim 19, wherein [[an]] the ambiguity creating location in the execution sequence is identified by a compiler.

32-36. (Cancelled)

37. (Previously Presented) The method of claim 19, wherein the particular operation comprises a memory referencing instruction.

38. (Previously Presented) The method of claim 37, wherein the memory referencing instruction comprises one or more of loads, stores and prefetches.

39-45. (Cancelled)

46. (Currently Amended) A method of preparing code for a processor, the method comprising:

preparing a tangible first executable instance of the code, the preparing identifying at least ambiguity creating locations therein;

executing the first executable instance and responsive to detection of an execution event, backtracking a displacement through the code to an operation thereof, wherein the

displacement is based, at least in part, on a type of operation appropriate to have triggered the execution event;

associating the operation with the execution event; and

classifying the execution event as ~~ambiguous or unambiguous~~ associated with one or more of the ambiguity creating locations or not associated with one or more of the ambiguity creating locations.

47. (Previously Presented) The method of claim 46, wherein the association between the associated operation and the execution event is based on a set of additional detections and responsive backtracking.

48. (Cancelled)

49. (Original) The method of claim 46, further comprising:

resolving at least some intervening ones of the identified ambiguity creating locations using branch history information.

50. (Currently Amended) A computer program product encoded in one or more computer readable media, the computer program product comprising:

an execution sequence of operations; and

one or more padding operations following a first operation of the execution sequence, the padding operations providing an unambiguous skid region of the execution sequence between the first operation and a subsequent ambiguity creating location to provide a displacement between the subsequent ambiguity creating location and the first operation[[s]] to cover an expected detection latency of the first operation.

51. (Previously Presented) The computer program of claim 50, wherein the first operation includes a memory access operation.

52. (Original) The computer program product of claim 50, wherein the padding operations include nops.

53. (Original) The computer program product of claim 50, wherein the unambiguous skid region does not include an ambiguity creating location.

54. (Previously Presented) The computer program product of claim 50, wherein the one or more computer readable media are selected from the set of a disk, tape, magnetic, optical, semiconductor or electronic storage medium.

55-58. (Cancelled)

59. (Currently Amended) An apparatus comprising:

means for identifying an ambiguity creating location;

means for backtracking a displacement ~~within an unambiguous skid region~~, from a point coinciding with an execution event in an execution sequence of operations on a processor, through the execution sequence toward a particular operation thereof that precedes the coinciding point, wherein the backtracking displacement is based, at least in part, on a type of the particular operation; [[and]]

means for determining that an ambiguity creating location is not encountered during the backtracking;

means for associating the execution event with the particular operation.

60. (Original) the apparatus of claim 59, further comprising:

means for bridging at least some ambiguity creating locations.

61. (Previously Presented) An apparatus comprising:

a code preparation facility suitable for preparation of an execution sequence of operations for a processor; and

means for padding the execution sequence to provide an unambiguous skid region between a particular operation and a subsequent ambiguity creating location within the sequence of operations to cover an expected detection latency, wherein the expected detection latency is based, at least in part, on a type of the particular operation.

62. (Previously Presented) The method of claim 3, wherein the ambiguity creating location is bridged using branch history information.

63. (Currently Amended) The method of claim 1 further comprising:

identifying the preceding operation and [[an]] the ambiguity creating location subsequent to the identified preceding operation;

determining that an unambiguous interval between the ambiguity creating location and the identified operation is insufficient to cover an expected detection latency for the identified preceding operation; and

inserting padding operations, which provide at least a portion of a skid region between the identified preceding operation and the ambiguity creating location, into the code subsequent to the identified operation, wherein the expected detection latency is based, at least in part, on a type of the identified preceding operation.

64. (Currently Amended) The method of claim 19 ~~further comprising identifying target operations and ambiguity creating locations in the sequence of operations~~, wherein the identified target operation[[s]] includes the particular operation.

65. (Currently Amended) The method of claim 64 further comprising ignoring a second execution event if [[an]] the ambiguity creating location is encountered while backtracking.

66. (Currently Amended) The method of claim 64 further comprising:

encountering [[an]] the ambiguity creating location while backtracking; and

bridging the ambiguity creating location using branch history information.

67. (Cancelled)

68. (Currently Amended) The method of claim [[67]] 64 further comprising inserting one or more padding operations into the sequence of operations between ~~a first of~~ the identified target operation[[s]] and ~~a first of~~ the ~~identified~~ ambiguity creating location[[s]] to cover an expected detection latency of the ~~first identified~~ target operation.

69. (Previously Presented) The method of claim 46, wherein the preparing comprises:

inserting one or more padding operations between the operation and a first of the identified ambiguity creating locations that is subsequent to the operation, wherein the one or more padding operations provide at least a portion of a skid region between the operation and the first identified ambiguity creating location sufficient to cover an expected detection latency of the operation.

REMARKS

This Amendment and Response is filed in reply to the final Office action dated April 18, 2007. Claims 1, 3, 5, 19, 31, 46, 59, 63, 65-66 and 68 are amended, claim 67 is canceled and claims 2, 4, 32-36, 39-45, 48 and 55-58 were previously cancelled. Accordingly, after entry of this Amendment and Response, claims 1, 3, 5-31, 37-38, 46-47, 49-54, 59-66 and 68-69 remain pending.

I. Objections to the Specification

The specification is objected to because it contains an embedded hyperlink and/or other form of browser-executable code. The Assignee respectfully disagrees. The specification, as previously amended, does not contain a hyperlink or other form of browser-executable code such as a URL placed between the symbols "<>" or http:// followed by a URL address. *See MPEP* § 608.01(VII). The Assignee respectfully requests that the objection to the specification be withdrawn.

II. Claim Rejections Under 35 U.S.C. § 112

Claims 3, 31, 46-47, 49, 60 and 62-69 are rejected under 35 U.S.C. § 112, first paragraph, as failing to comply with the written description requirement. Specifically, in our last response claims 1, 19 and 59 were amended to include the limitation of "backtracking a displacement within an unambiguous skid region." The Office action alleges that the backtracking occurs within unambiguous skid region that does not contain an ambiguity creating location. However, claims 3, 31, 60 and 63-69, which depend from one of these claims, recite the limitation of an ambiguity creating location. The Office action alleges that the specification does not have any disclosure describing an unambiguous skid region containing an ambiguity creating location. *See Office action*, page 3. Without admitting to any deficiency in the specification or claims, we have elected to clarify claims 1, 19 and 56 to remove "within an unambiguous skid region" from the backtracking limitation.

Claim 46 was amended in our last response to recite the limitation of "classifying the execution event as ambiguous or unambiguous." The Office action alleges that the specification has no disclosure of classifying an event as ambiguous or unambiguous. *See Id.* The Assignee respectfully submits that an execution event may be classified as unambiguous when the backtracking does not encounter an ambiguity creating location, and as ambiguous when an ambiguity creating location is encountered during the backtracking. That is, the presence or absence of execution path ambiguity while backtracking determines whether or not the execution event is ambiguous. *See Specification*, paragraphs 22-23. Claims 47 and 49, which depend from claim 46, suffer the same deficiency.

Nonetheless, claim 46 is amended to recite the limitation "classifying the execution event as associated with one or more of the ambiguity creating locations or not associated with one or more of the ambiguity creating locations." Support for this claim amendment may be found at least at paragraph 22 and Fig. 2 of the specification. It is respectfully submitted that claims 3, 31, 46-47, 49, 60 and 62-69 are now in compliance with 35 U.S.C. § 112, first paragraph, and such indication is respectfully requested.

Claims 1, 3, 5-31, 37-38, 59-60 and 62-69 are rejected under 35 U.S.C. § 112, second paragraph, as failing to set forth the subject matter which applicant(s) regard as their invention. The claims, as amended, particularly point out and distinctly claim the subject matter which the applicant regards as his invention. While the Office action alleges a narrow purpose for the invention, the purpose of the invention is broader than this narrow interpretation, as set forth in the claims.

III. Claim Rejections Under 35 U.S.C. § 102

Claims 1, 3, 5-15, 19-31, 37-38, 46-47, 49, 59-60 and 62 are rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,964,867 to Anderson et al. (hereinafter "Anderson). An anticipation rejection requires that each and every limitation of a claim be disclosed in a single prior art reference.

Initially, the rejection of independent claims 1, 19, 46 and 59 is addressed. Claim 1 includes a limitation "identifying an ambiguity creating location in the code, the ambiguity creating location being an operation reachable from a plurality of program execution paths." Claims 19, 46 and 59 include a similar limitation. The Office action alleges that Anderson teaches this by stating the analysis can identify execution paths. *See Office action*, page 13. The Assignee respectfully disagrees. In Anderson, when a branch instruction is encountered, branch history bits are used to try and resolve whether or not the branch was taken (*see Anderson*, column 24, lines 15-31), rather than identifying an ambiguity creating location as required by the independent claims 1, 19 and 46. Further, independent claims 1 and 19 are amended to include a limitation "determining that the ambiguity creating location is not encountered while backtracking." Claim 59, as amended, includes a similar limitation. It is respectfully submitted that Anderson does not teach such a limitation.

Independent claim 46, as amended, includes the limitation "classifying the execution event as associated with one or more of the ambiguity creating locations or not associated with one or more of the ambiguity creating locations." It is respectfully submitted that Anderson does not teach such a limitation.

Insofar as Anderson does not teach all of the limitations of independent claims 1, 19, 46 and 59, it cannot anticipate them. Therefore, it is respectfully submitted that independent claims 1, 19, 46 and 59 are patentable over Anderson, believed to be in form for allowance,

and such indication is respectfully requested. The remaining claims 3, 5-15, 20-31, 37-38, 47, 49, 60 and 62 all depend, either directly or indirectly, from one of independent claims 1, 19, 46 and 59. Accordingly, these dependent claims are themselves patentable over Anderson for at least the reasons set forth above and such indication is respectfully requested. This statement is made without reference to or waiving the independent bases of patentability within each dependent claim.

IV. Claim Rejections Under 35 U.S.C. § 103

Claims 16-18 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Anderson in view of U.S. Patent Publication No. 2002/0010913 to Ronstrom (hereinafter "Ronstrom"). Claims 63-68 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Anderson in view of V. Bala and N. Rubin, "Efficient Instruction Scheduling Using Finite State Automata," *Proceedings of MICRO-28*, IEEE 1995, pp. 46-56 (hereinafter "Bala"). Claims 50-54 and 61 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Ronstrom in view of Bala.

A. Rejection of Claims 16-18

Claims 16-18 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Anderson in view of Ronstrom. This rejection is respectfully traversed. Claims 16-18 depend from independent claim 1. As discussed above, claim 1 is patentable over Anderson. Further, the Office action only relies upon Ronstrom to teach a compiler that generates dummy instruction code for lowering cache miss penalty and inserting the same. *See Office action*, page 15. As such, Ronstrom is insufficient to remedy the deficiency of Anderson regarding identifying an ambiguity creating location, as required by amended claim 1. Accordingly, withdrawal of the rejection is respectfully requested.

B. Rejection of Claims 63-68

Claims 63-68 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Anderson in view of Bala. Claims 63-68 depend from one of independent claims 1 and 19. As discussed above, claims 1 and 19 are patentable over Anderson. The Office action relies on Bala to teach identifying an ambiguity creating location. *See Office action*, page 17. The Assignee respectfully disagrees that Bala teaches such a limitation. Bala teaches insertion of instructions to prevent a structural hazard. *See Bala*, Introduction Section, first paragraph. Bala also teaches placement of compensation code that gets generated when an instruction is speculatively executed above a branch upon which it is control dependent. *See Bala*, Introduction Section, paragraph nine. However, Bala fails to teach identifying ambiguity creating locations which are those instructions which may be reached from a plurality of code execution paths as required by independent claims 1 and 19. Therefore, the Assignee respectfully submits that independent claims 1 and 19 are patentable over Anderson in view

of Bala and such indication is respectfully requested. Rejected claims 63-68 depend from, either directly or indirectly, from one of independent claims 1 and 19. Accordingly, these dependent claims are themselves patentable over Anderson in view of Bala for at least the reasons set forth and such indication is respectfully requested.

C. Rejection of Claims 50-54 and 61

Claims 50-54 and 61 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Ronstrom in view of Bala. The Assignee respectfully traverses this rejection. Initially, the rejection of independent claims 50 and 61 is addressed.

The Examiner asserts that a branch is an ambiguity creating location. See *Office action*, page 23. The Assignee respectfully disagrees. An ambiguity creating location is a location reachable from a plurality of program execution paths. As discussed above, Bala does not contemplate ambiguity creating locations. Further, Ronstrom is insufficient to remedy the deficiency of Bala. Therefore, the Assignee respectfully submits that independent claims 50 and 61 are patentable over Ronstrom in view of Bala and such indication is respectfully requested. The remaining rejected claims 51-54 all depend, either directly or indirectly, from independent claim 50. Accordingly, these dependent claims are themselves patentable over Ronstrom in view of Bala for at least the reasons set forth above and such indication is respectfully requested.

V. Conclusion

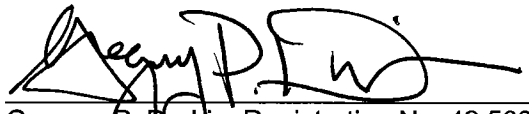
The Applicant thanks the Examiner for his thorough review of the application. The Applicant respectfully submits the present application, as amended, is in condition for allowance and respectfully requests the issuance of a Notice of Allowability as soon as practicable.

This Amendment is submitted contemporaneously with a Request for Continued Examination and a petition for a one-month extension of time in accordance with 37 C.F.R. § 1.136(a). Accordingly, please charge Deposit Account No. 04-1415 in the amount of $910.00 ($790.00 for the Request for Continued Examination fee and $120.00 for a one-month extension of time fee). The Applicant believes no further fees or petitions are required. However, if any such petitions or fees are necessary, please consider this a request therefor and authorization to charge Deposit Account No. 04-1415 accordingly.

If the Examiner should require any additional information or amendment, please contact the undersigned attorney.

Dated: August 20, 2007

Respectfully submitted,



Gregory P. Durbin, Registration No. 42,503
Attorney for Applicant
USPTO Customer No. 66083

DORSEY & WHITNEY LLP
Republic Plaza Building, Suite 4700
370 Seventeenth Street
Denver, Colorado 80202-5647
Phone: (303) 629-3400
Fax: (303) 629-3450